gxformat2 Documentation

Release 0.18.0

Galaxy Project and Community

May 12, 2023

Contents

Contents:

CHAPTER 1

Format 2

This package defines a high-level Galaxy workflow description termed "Format 2". The current schema version is v19_09 and the schema can be found here. This version of workflow format can be consumed by Galaxy since version 19.09.

The Format 2 workflow description is still somewhat experimental and may yet change in small potentially backward incompatible ways until the format is exported by Galaxy by default.

The traditional Galaxy workflow description (files ending in `.ga` extension, sometimes called native workflows in this project) was not designed to be concise and is neither readily human readable or human writable. Galaxy workflow Format 2 is being designed to address these limitations, while also moving Galaxy's workflow description language toward standards such as the Common Workflow Language.

CHAPTER 2

gxformat2

This Python project can be installed from PyPI using `pip`.

```
$ pip install gxformat2
```

Checkout the project tests or how it used in projects such as Planemo and Galaxy to see how to use the gxformat2 library. Reference documentation for the modules can be found as part of the project's documentation.

This project also includes various scripts for working with Galaxy workflows. Checkout their help for more information.

```
$ gxwf-lint --help
$ gxwf-viz --help
$ gxwf-abstract-export --help
```

This library and associated scripts are licensed under the MIT License.

CHAPTER 3

Installation

```
$ pip install gxformat2
```

CHAPTER 4

gxformat2 Project Code of Conduct

This code of conduct outlines our expectations for participants within the Galaxy community, as well as steps to reporting unacceptable behavior. We are committed to providing a welcoming and inspiring community for all and expect our code of conduct to be honored. Anyone who violates this code of conduct may be banned from the community.

Our open source community strives to:

- **Be friendly and patient.**
- **Be welcoming**: We strive to be a community that welcomes and supports people of all backgrounds and identities. This includes, but is not limited to members of any race, ethnicity, culture, national origin, colour, immigration status, social and economic class, educational level, sex, sexual orientation, gender identity and expression, age, size, family status, political belief, religion, and mental and physical ability.
- **Be considerate**: Your work will be used by other people, and you in turn will depend on the work of others. Any decision you take will affect users and colleagues, and you should take those consequences into account when making decisions. Remember that we're a world-wide community, so you might not be communicating in someone else's primary language.
- **Be respectful**: Not all of us will agree all the time, but disagreement is no excuse for poor behavior and poor manners. We might all experience some frustration now and then, but we cannot allow that frustration to turn into a personal attack. It's important to remember that a community where people feel uncomfortable or threatened is not a productive one.
- **Be careful in the words that we choose**: We are a community of professionals, and we conduct ourselves professionally. Be kind to others. Do not insult or put down other participants. Harassment and other exclusionary behavior aren't acceptable. This includes, but is not limited to: Violent threats or language directed against another person, Discriminatory jokes and language, Posting sexually explicit or violent material, Posting (or threatening to post) other people's personally identifying information ("doxing"), Personal insults, especially those using racist or sexist terms, Unwelcome sexual attention, Advocating for, or encouraging, any of the above behavior, Repeated harassment of others. In general, if someone asks you to stop, then stop.
- **Try to understand why we disagree**: Disagreements, both social and technical, happen all the time. It is important that we resolve disagreements and differing views constructively. Remember that we're different. Diversity contributes to the strength of our community, which is composed of people from a wide range of backgrounds. Different people have different perspectives on issues. Being unable to understand why someone

holds a viewpoint doesn’t mean that they’re wrong. Don’t forget that it is human to err and blaming each other doesn’t get us anywhere. Instead, focus on helping to resolve issues and learning from mistakes.

Diversity Statement

We encourage everyone to participate and are committed to building a community for all. Although we will fail at times, we seek to treat everyone both as fairly and equally as possible. Whenever a participant has made a mistake, we expect them to take responsibility for it. If someone has been harmed or offended, it is our responsibility to listen carefully and respectfully, and do our best to right the wrong.

Although this list cannot be exhaustive, we explicitly honor diversity in age, gender, gender identity or expression, culture, ethnicity, language, national origin, political beliefs, profession, race, religion, sexual orientation, socioeconomic status, and technical ability. We will not tolerate discrimination based on any of the protected characteristics above, including participants with disabilities.

Reporting Issues

If you experience or witness unacceptable behavior, or have any other concerns, please report it to any combination of the following people that makes you feel the most comfortable:

- Dave Clements (clementsgalaxy@gmail.com). Dave is the Galaxy Project community outreach manager and has experience handling Code of Conduct related issues.
- Dr. Mike Schatz (mschatz@cs.jhu.edu). Mike is Dave Clements’ direct manager and issues related to Dave in some way should be reported to Mike.
- Helena Rasche (helena.rasche@gmail.com). Helena is a well-known, trusted community member, is LGBT+, and has completely separate funding and institutional affiliation from Dave and Mike.

All reports will be handled with discretion. In your report please include:

- Your contact information.
- Names (real, nicknames, or pseudonyms) of any individuals involved. If there are additional witnesses, please include them as well. Your account of what occurred, and if you believe the incident is ongoing. If there is a publicly available record (e.g. a mailing list archive or a public IRC logger), please include a link.
- Any additional information that may be helpful.

After filing a report, a representative will contact you personally, review the incident, follow up with any additional questions, and make a decision as to how to respond. If the person who is harassing you is part of the response team, they will recuse themselves from handling your incident. If the complaint originates from a member of the response team, it will be handled by a different member of the response team. We will respect confidentiality requests for the purpose of protecting victims of abuse.

Attribution & Acknowledgements

This code of conduct is based on the Open Code of Conduct from the TODOGroup.

CHAPTER 5

Contributing

Please note that this project is released with a Contributor Code of Conduct. By participating in this project you agree to abide by its terms.

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

5.1 Types of Contributions

5.1.1 Report Bugs

Report bugs at https://github.com/galaxyproject/gxformat2/issues.

If you are reporting a bug, please include:

- Your operating system name and version, versions of other relevant software such as Galaxy or Planemo.
- Links to relevant tools and workflows.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

5.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” is open to whoever wants to implement it.

5.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” is open to whoever wants to implement it.

5.1.4 Write Documentation

gxformat2 could always use more documentation, whether as part of the official gxformat2 docs, in docstrings, or even on the web in blog posts, articles, and such.

5.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/galaxyproject/gxformat2/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- This will hopefully become a community-driven project and contributions are welcome :)

5.2 Get Started!

Ready to contribute? Here's how to set up *gxformat* for local development.

1. Fork the *gxformat2* repo on GitHub.
2. Clone your fork locally:

   ```
   $ git clone git@github.com:your_name_here/gxformat2.git
   ```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

   ```
   $ make setup-venv
   ```

4. Create a branch for local development:

   ```
   $ git checkout -b name-of-your-bugfix-or-feature
   ```

   Now you can make your changes locally.

5. When you're done making changes, check that your changes pass `flake8` and the tests

   ```
   $ tox -e lint
   $ tox -e mypy
   $ tox -e unit
   ```

6. Commit your changes and push your branch to GitHub:

   ```
   $ git add .
   $ git commit -m "Your detailed description of your changes."
   $ git push origin name-of-your-bugfix-or-feature
   ```

7. Submit a pull request through the GitHub website.

5.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring.
2. The pull request should work for Python >=3.6. Check CI results on pull request and make sure that the tests pass for all supported Python versions.

5.3.1 Tox

Tox is a tool to automate testing across different Python versions. The `tox` executable can be supplied with a `-e` argument to specify a testing environment. gxformat2 defines the following environments:

`py37-lint` Lint the gxformat2 code using Python 3.7.

`py37-lint_docstrings` Lint the project Python docstrings.

`py36-unit` Run the unit tests with Python 3.6.

CHAPTER 6

Release Checklist

This page describes the process of releasing new versions of gxformat2.

This release checklist is based on the Pocoo Release Management Workflow.

This assumes `~/.pypirc` file exists with the following fields (variations) are fine.

```
[distutils]
index-servers =
    pypi
    test

[pypi]
username:<username>
password:<password>

[test]
repository:https://testpypi.python.org/pypi
username:<username>
password:<password>
```

- Review `git status` for missing files.
- Verify the latest Travis CI builds pass.
- `make open-docs` and review changelog.
- Ensure the target release is set correctly in `galaxy/__init__.py` (`version` will be a `devN` variant of target release).
- `make clean && make lint && make test`
- `make release`

  This process will push packages to test PyPI, allow review, publish to production PyPI, tag the git repository, push the tag upstream, and modify the Homebrew recipe. If changes are needed, such as manual changes to the homebrew recipe, this can be broken down into steps such as:

  - `make release-local`

  - `make push-release`
  - `make release-brew`

CHAPTER 7

History

7.1 0.18.0 (2023-05-12)

- Fix input conversion if input has no label by @mvdbeek in https://github.com/galaxyproject/gxformat2/pull/89

7.2 0.17.0 (2023-01-06)

- Enable "when" for workflow steps by @mr-c in https://github.com/galaxyproject/gxformat2/pull/74
- When fixes by @mvdbeek in https://github.com/galaxyproject/gxformat2/pull/86

7.3 0.16.0 (2022-09-20)

- Add dev `when` on steps to backend (don't expose in schema yet). by @jmchilton in https://github.com/galaxyproject/gxformat2/pull/48
- Update project plumbing to allow dev release. by @jmchilton in https://github.com/galaxyproject/gxformat2/pull/49
- Drop support for Python 3.5, add 3.9 by @nsoranzo in https://github.com/galaxyproject/gxformat2/pull/52
- Relicense under the MIT license by @nsoranzo in https://github.com/galaxyproject/gxformat2/pull/58
- Format2: Add *label* attribute to *WorkflowInputParameter* and *WorkflowOutputParameter* by @nsoranzo in https://github.com/galaxyproject/gxformat2/pull/56
- Misc fixes and refactorings by @nsoranzo in https://github.com/galaxyproject/gxformat2/pull/55
- Convert Format2 workflow *label* to native *name* by @nsoranzo in https://github.com/galaxyproject/gxformat2/pull/54
- test_abstract_export: use different names for the different outputs by @simleo in https://github.com/galaxyproject/gxformat2/pull/57

- Fix 2 typos by @nsoranzo in https://github.com/galaxyproject/gxformat2/pull/62
- Propagate *doc* field to abstract CWL format by @nsoranzo in https://github.com/galaxyproject/gxformat2/pull/65
- Linting fixes by @mvdbeek in https://github.com/galaxyproject/gxformat2/pull/64
- Maintain collection_type if present by @mvdbeek in https://github.com/galaxyproject/gxformat2/pull/68
- Fix schema doc build by @nsoranzo in https://github.com/galaxyproject/gxformat2/pull/69
- Lint and deprecation fixes by @nsoranzo in https://github.com/galaxyproject/gxformat2/pull/70
- Run java codegenerator by @mvdbeek in https://github.com/galaxyproject/gxformat2/pull/71
- Run maven tests on pull_request by @mvdbeek in https://github.com/galaxyproject/gxformat2/pull/72
- fix schema-salad pycodegen by @mr-c in https://github.com/galaxyproject/gxformat2/pull/76
- Add workflow default file support by @mvdbeek in https://github.com/galaxyproject/gxformat2/pull/79
- Add typescript implementation by @mr-c in https://github.com/galaxyproject/gxformat2/pull/75
- Fix cytoscape HTML exports from dist package. by @jmchilton in https://github.com/galaxyproject/gxformat2/pull/82
- Add missing elements to schema, fix change_datatype conversion, CSS by @mvdbeek in https://github.com/galaxyproject/gxformat2/pull/83
- Support lists as data inputs by @mvdbeek in https://github.com/galaxyproject/gxformat2/pull/84

7.4 0.15.0 (2020-08-12)

- Lint types of default values.
- Fix bugs in schema related to differing type names between Galaxy and CWL.
- Generate cwl v1.2 instead of cwl v1.2.0-dev5 release now that it has been released.
- More testing of linting and CWL 1.2 export.

7.5 0.14.0 (2020-08-11)

- Bug fix where native export had explicit outputs declaration still in it (wouldn't break anything, but was deceptive).
- Fixes for experimental CWL 1.2 abstract export.
- Improve script structures and documentation.
- Improve code structure - add more types, make more things immutable, mention mutability in docstrings.

7.6 0.13.1 (2020-08-03)

- Improve package structure - publish fixed sphinx docs, fix readme badges, add mypy typing support.

7.7 0.13.0 (2020-07-30)

- Add experimental export to CWL 1.2 using new abstract Operation classes.

7.8 0.12.0 (2020-07-27)

- Drop support for Python 2 - to support next bullet.
- Update schema parser for recent changes to schema salad.

7.9 0.11.4 (2020-07-27)

- Added abstraction for uniform access to workflow outputs across formats.

7.10 0.11.3 (2020-07-23)

- Bug fixes for exporting newer input concepts from native to Format 2.
- Added abstraction for uniform access to workflow inputs across formats.

7.11 0.11.2 (2020-07-22)

- Rework cytoscape and helpers for reuse from Planemo.
- Rev markdown validator for and from latest Galaxy changes.

7.12 0.11.1 (2020-02-25)

- Bug fix for gxwf-lint invocation from setup.py installed script.

7.13 0.11.0 (2020-02-25)

- Validate Galaxy Markdown in workflow reports as part of linting.
- Improved null handling in native ga workflow linting.
- Enhancements to workflow linting from Python. Lint for lack of documentation, tools using the test toolshed, and implement special linting for training material workflows to ensure a tag matches the workflow topic.
- Add gxwf-viz script that produces a cytoscape visualization of a workflow.

7.14 0.10.1 (2019-12-07)

- Bug fix to handle outputs without labels in Format 2 - they don't validate per se but they are important for testing in the Galaxy framework.

7.15 0.10.0 (2019-12-06)

- Implement scheam, validation, linting (for Format 2 and .ga).
- Handle new reports field in Galaxy 19.09 workflows.
- Numerous fixes for conversiion to and from native workflows.
- Numerous new test cases.
- Implement Java project for valiating and linting both kinds of workflows.

7.16 0.9.0 (2019-07-08)

- Implement default values in gxformat2.

7.17 0.8.4 (2019-06-24)

- Fix output IDs of 0.

7.18 0.8.3 (2019-05-23)

- Implement set_columns PJA.

7.19 0.8.2 (2019-03-16)

- Allow another API return option for experimental tool creation API.

7.20 0.8.1 (2019-03-11)

- Implement change datatype PJA.

7.21 0.8.0 (2018-11-01)

- Implement experimental CWL-style step defaults (see Galaxy PR #6850).

7.22 0.7.1 (2018-10-09)

- Various small fixes for changes in 0.7.1.

7.23 0.7.0 (2018-10-08)

- Add some basic test cases.
- Allow ID-map style listing of steps.
- Ordered load (in addition to existing dump functionality) or ordering of steps in ID-map style variant works.
- Allow CWL-style $graph defs that can define multiple workflows in a single file.
- Initial work on de-duplicating subworkflow definitions on import.
- Fix position handling while exporting workflow.

7.24 0.6.1 (2018-10-01)

- Fix export of non-data parameters and implicit workflow connections.

7.25 0.6.0 (2018-10-01)

- Various fixes, allow id map style workflow input definitions.

7.26 0.5.0 (2018-10-01)

- More fixes for PJA, add the `doc` keyword to format 2 workflows to match CWL workflows. Map to and from native Galaxy workflows as annotations.

7.27 0.4.0 (2018-10-01)

- Fixes for exporting PJA when exporting workflows from native .ga to format 2.

7.28 0.3.2 (2018-10-01)

- Fixes for exporting workflow outputs from native .ga to format 2, support for modern map style output definitions like CWL 1.0.

7.29 0.3.1 (2018-10-01)

- Fixes for exporting subworkflows from native .ga to format 2.

7.30 0.3.0 (2018-09-30)

- More cwl style inputs, initial work on conversion from native workflows, various small fixes and tweaks.

7.31 0.2.0 (2018-02-21)

- Bring in latest Galaxy updates - Python 3 fixes, safe YAML usage, and more PJA implemented.

7.32 0.1.1 (2016-08-15)

- Fix one Python 3 incompatibility.

7.33 0.1.0 (2016-05-02)

- Initial version - code from Galaxy's test framework with changes based on planemo testing.

CHAPTER 8

Indices and tables

- genindex
- modindex
- search